SCENARIO

The application is vulnerable to web cache poisoning because it excludes a certain parameter from the cache key. There is also inconsistent parameter parsing between the cache and the back-end. We will try to poison the cache with a response that executes alert(1) in the visitor's browser.

**PROCEDURE**

1. Open the web application and in the BurpSuite’s Proxy tab send the GET request for **/js/geolocate.js?callback=setCountryCookie** to BurpSuite’s Repeater and study it.
2. Using the **Param Miner’s Rails parameter cloaking scan** we see that the **utm\_content** parameter is supported and is also excluded from the cache key.
3. Notice that if we use a semicolon (;) to append another parameter to utm\_content, the cache treats this as a single parameter. This means that the extra parameter is also excluded from the cache key.
4. Observe that every page imports the script **/js/geolocate.js**, executing the callback function **setCountryCookie()**.
5. Notice that we can control the name of the function that is called on the returned data by editing the URL’s callback parameter. However, we can't poison the cache for other users in this way because the parameter is keyed.
6. Study the cache behavior. Observe that if we add duplicate callback parameters, only the final one is reflected in the response, but both are still keyed. However, if we append the second callback parameter to the **utm\_content** parameter using a semicolon, it is excluded from the cache key and still overwrites the callback function in the response.
7. According to the way our injected query parameter we will craft an exploit string which will break out of that tag and trigger our alert by executing arbitrary JavaScript.
8. Append the crafted exploit as shown in the Payload in the URL.
9. Send the malicious request after removing the cache buster parameter and keep replaying the request until we see our exploit server URL being reflected in the response and **X-Cache: hit** in the headers.

**PAYLOAD**

/js/geolocate.js?callback=setCountryCookie&utm\_content=poison;callback=alert(1)

**REMEDIATION**